Proposal to

THE ADVANCED RESEARCH PROJECTS AGENCY

For continuation of

THE HEURISTIC PROGRAMMING PROJECT

ARPA Order No. 2494

February 1977

Computer Science Department
STANFORD UNIVERSITY

Stanford University February, 1977

Proposal to

THE ADVANCED RESEARCH PROJECTS AGENCY

for continuation of

THE HEURISTIC PROGRAMMING PROJECT

Edward A. Feigenbaum, Professor of Computer Science
Joshua Lederberg, Professor of Genetics
Bruce G. Buchanan, Adjunct Professor of Computer Science

Co-Principal Investigators

Abstract

The research project in Heuristic Programming is proposed for continuation of Contract DAHC-15-73-C-0435 for two years beginning July 1, 1976 at a total cost of $650,000.00. This project will be administratively distinct within the Computer Science Department.

STANFORD UNIVERSITY HEURISTIC PROGRAMMING PROJECT

Principal Investigators
Prof. Edward A. Feigenbaum
Prof. Joshua Lederberg
Prof. Bruce G. Buchanan

Contract No. DAHC 15-73C-0435
July 1, 1977 to June 30, 1979

1 OBJECTIVES

The objectives of the Heuristic Programming project are to provide conceptual and programming tools for building knowledge-based programs. More specifically the proposed work centers on three major themes:

I-A. Generalization of knowledge-based systems design and implementation techniques.

I-B. Extension of current research toward new or more powerful techniques for knowledge-based systems.

II. Orientation toward other ARPA-IPTO advanced R&D efforts.

2 BACKGROUND AND TECHNICAL NEED

2.1 Symbolic Computation

Computer scientists have long recognized that a computer is a general symbol-manipulating device. Arithmetic constitutes a special case of this capability--the manipulation of those symbols that are numbers. In this proposal we will be discussing non-numeric symbol manipulation by computers. In thinking about non-numeric computation, it is useful to think about:

a. inference methods (as opposed to calculation and algorithms)

b. qualitative "lines of reasoning" (as opposed to quantitative formulations)

c. symbolic facts (not merely numeric parameters and formulas)

d. decision rules of expertise and judgment (as opposed to mathe-matical decision rules)

Symbolic computation, though general and powerful, has hardly begun to be exploited in real applications. The specialty within Computer Science that has studied complex methods of symbolic computation is "Artificial Intelligence Research".

2.2 The Intelligent Agent Viewpoint

Though artificial intelligence research has a number of different goals, one of them can be described as the study, design and implementation of "intelligent agent" computer programs. A succinct version of this view was presented by Feigenbaum in a report to the Director of ARPA in 1973, as follows:

Artificial Intelligence research is that part of Computer Science that is concerned with the symbol-manipulation processes that produce intelligent action. By "intelligent action" is meant an act or decision that is goal-oriented, arrived at by an understandable chain of symbolic analysis and reasoning steps, and is one in which knowledge of the world informs and guides the reasoning. The potential uses of computers by people to accomplish tasks can be "one-dimensionalized" into a spectrum representing the nature of instruction that must be given the computer to do its job. Call it the WHAT-TO-HOW spectrum. At one extreme of the spectrum, the user supplies his intelligence to instruct the machine with precision exactly HOW to do his job, step-by-step. Progress in Computer Science can be seen as steps away from that extreme "HOW" point on the spectrum: the familiar panoply of assembly languages, subroutine libraries, compilers, extensible languages, etc. At the other extreme of the spectrum is the user with his real problem (WHAT he wishes the computer, as his instrument, to do for him). He aspires to communicate WHAT he wants done in a language that is comfortable to him (perhaps English); via communication modes that are convenient for him (including perhaps, speech or pictures); with some generality, some abstractness, perhaps some vagueness, imprecision, even error; without having to lay out in detail all necessary subgoals for adequate performance - with reasonable assurance that he is addressing an intelligent agent that is using knowledge of his world to understand his intent, to fill in his vagueness, to make specific his abstractions, to correct his errors, to discover appropriate subgoals, and ultimately to translate WHAT he really wants done into processing steps that define HOW it shall be done by a real computer. The research activity aimed at creating computer programs that act as "intelligent agents" near the WHAT end of the WHAT-TO-HOW spectrum can be viewed as the long-range goal of AI research.

Thus, the Intelligent Agent is a knowledge-based system--the essential system component is a body of knowledge representing the user's problem domain.

2.3 Knowledge-based Systems Research and Design

Early work in artificial intelligence aimed toward the creation of generalized problem solvers. Work on programs like GPS [by Newell and Simon] and theorem proving [Nilsson71], for instance, was inspired by the apparent generality of human intelligence and motivated by the belief that it might prove possible to develop a single program applicable to all (or most) problems. While this early work demonstrated that there was a large body of useful general purpose techniques (such as problem decomposition into subgoals, and heuristic search in its many forms), these techniques did not by themselves offer sufficient power for expert levels of performance. Recent work has instead focused on the incorporation of large amounts of task specific knowledge in what have been called "knowledge-based" systems. Rather than non-specific problem solving power, knowledge based systems have emphasized high performance based on the accumulation of large amounts of knowledge about a single domain. A second successful focus in work on intelligent systems has been the emphasis on the utility of solving "real world" problems, rather than artificial problems fabricated in simplified domains. This is motivated by the belief that artificial problems may prove in the long run to be more a diversion than a foundation for further work, and by the belief that the field has developed sufficiently to provide techniques that can aid working scientists. While artificial problems may serve to isolate and illustrate selected aspects of a task, solutions developed for those selected aspects often do not generalize well to the complete problem.

There are numerous current examples of successful systems embodying both of these trends, systems which apply task-specific knowledge to real world problems.

Our project is widely regarded as the initiator of this line of endeavor. Over the past 12 years, with support from ARPA, NIH, and NSF, our scientists have studied extensively the problems of knowledge-based systems design and have implemented (or are in the process of implementing) a number of such systems. The accomplishments are summarized in the proposal section on Accomplishments, but a narrative here may prove useful.

DENDRAL: An intelligent assistant to an analytic and structural chemist. It infers the structures of complex organic molecules from structural constraints. These constraints are either supplied interactively by the user from his "private"

knowledge and intuition, or are inferred automatically from instrument data, such as mass spectral data, nuclear magnetic resonance data,etc. For those families of molecules for which the knowledge base has been carefully elaborated, the DENDRAL program performs at levels equalling or exceeding the best human experts. The DENDRAL program now has a significant user community in university laboratories and in industry, and is being used to solve difficult real problems.

Meta-DENDRAL: This program is focused on the problem of elaborating DENDRAL's knowledge base for specific families of compounds. It infers an empirical theory (a body of fragmentation rules) of the mass spectrometry of specific families from recorded mass spectral data. It has not only "rediscovered" rules previously acquired from chemists, but has discovered novel rules for certain families--rules that have recently warranted publication in the chemical literature.

MYCIN: This program is an intelligent assistant to a physician diagnosing infectious diseases. In conjunction with its diagnoses, it recommends theraputic action. It is capable of explaining its line-of-reasoning in any (and varying) level of detail to the user in English. It can accept new decision rules from the user in English. It keeps an updated model of its own knowledge base, which it uses to critique the introduction of new rules into the system. It is capable of acquiring and using measures of the uncertainty of the knowledge, and produces a "believability" index with each inference,i.e. it is capable of approximate implication. A version called EMYCIN, sans infectious disease knowledge, has been developed to extend the use of the system to other domains. (One of the project scientists, E. Shortliffe, was recently awarded the Grace M. Hopper award of the Association for Computing Machinery for his distinguished work on MYCIN).

HASP: Project scientists working in a classified environment led the development of a signal-understanding program for continuous surveillance of certain objects of military interest. The program ran successfully in a number of highly varied test situations, and is being further developed in a currently-funded ARPA program. The program used a design for incremental hypothesis formation that was a modification of the HEARSAY design for the CMU speech-understanding system. Symbolic knowledge from a number of sources was used to aid the interpretation of the primary signal data. Time-dependent analysis was novel in this system and played an important role.

AM: This remarkable program conjectures "interesting" mathematical concepts. Its knowledge base encompasses the (usually private) knowledge of a mathematician as to what constitutes an "interesting" construct in mathematics. Starting with the simplest set-theory concepts, and hundreds of rules

defining "interestingness" of mathematical concepts, it has conjectured such concepts as addition, multiplication, factorization, primes, unique factorization into primes (the fundamental theorem of arithmetic), and an almost unstudied concept in number theory called "maximally divisible numbers".

MOLGEN: (under development) This program is being designed to be an intelligent assistant to an experimental molecular geneticist in formulating plans for laboratory experiments involving the manipulation of short DNA strands with restriction enzymes. The program is concerned with representing knowledge about planning and with the automatic formulation of plans to the level of detail demanded by the user. The program's knowledge must be represented at various levels--biological,genetic,topological, and chemical--and these levels must be incorporated into the reasoning.

Crystallographic Image Interpretation: (under development) This program is being designed to interpret ambiguous, incomplete three-dimensional image data obtained in x-ray crystallography of protein structures. The image input data is the so-called electron density map and the answer desired is an approximately correct protein molecule (or portion thereof). As with HASP, many sources of symbolic data support the interpretation of the primary signal data. The HASP program organization has been imported into this program as a test of its generality. The interpretation problem is difficult because the best wavelength available (x-rays) is too long to resolve atoms and interatomic separations;hence the need for additional sources of symbolic knowledge,e.g. the amino acid sequence of the protein.

In short, as the capsule sketches above indicate, the main themes of our work involve: the acquisition and maintenance of knowledge bases; the utilization of this knowledge in a variety of ways for data interpretation, problem solving, and planning; and the representation of this knowledge for computer inference. These programs constitute superb "laboratories" in which to study the problems of design of knowledge-based systems. Though built to a large extent with non-ARPA support, they serve well the purposes of the ongoing and proposed ARPA-sponsored research.

2.4 Production-rule-based Systems: Methodology and System Features

We sketch below our approach to issues of knowledge representation, management and use. It consists of (a) a representation of knowledge in terms of production rules and (b) interactive programs for acquiring, using, and explaining the knowledge in the system. An example of a production rule from the MYCIN domain is shown below, in both the internal format

(LISP), and the English translation which is generated from it. pplicable approach to knowledge representation, management and use. That foundation consists of (a) a representation of knowledge in terms of production rules and (b) interactive programs for acquiring, using, and explaining the knowledge in the system. An example of a production rule from the domain of infectious disease diagnosis is shown below, in both the internal format (LISP), and the English translation which is generated from it.

```
PREMISE   ($AND (SAME CNTXT GRAM GRAMNEG)
                (SAME CNTXT MORPH ROD)
                (SAME CNTXT AIR ANAEROBC))

ACTION    (CONCLUDE CNTXT IDENT BACTEROIDES TALLY .6)
```

If 1) the gram stain of the organism is gram negative, and
2) the morphology of the organism is rod, and
3) the aerobicity of the organism is anaerobic,
then there is suggestive evidence (.6) that the identity of the organism is Bacteroides.

The premise of the rule is a Boolean combination of preconditions, each of which is a predicate testing the value of an associative triple, and is thus value triple.

Such rules contain modular "chunks" of knowledge about the domain, represented in a form that will be comprehensible to someone who is acquainted with the field, but may know nothing about computer programming. Many unique and important capabilities are made possible by encoding knowledge in this form, capabilities reflected in the organization and architecture of the system.

The knowledge base of a system contains the collection of domain specific rules which give the system its competence in a particular domain. The problem date base contains information about a specific problem given to the system.

An interactive "intelligent agent" program uses the collection of rules in the knowledge base to reason about the task at hand. If, for, instance, the MYCIN program is attempting to determine the identity of an organism responsible for a particular infection, it retrieves the entire list of rules which, like the one above, conclude about identity. It then attempts to ascertain whether the conclusion of the first rule is valid, by evaluating in turn each of the clauses of the premise. Thus, for the rule above, the first thing to find out is the gram stain of the current organism. If this information is already

available in the data base, the program retrieves it. If not, determination of gram stain becomes the objective of a new rule, and the program retrieves all rules which conclude about it, and tries to use each of them to obtain the value of gram stain. If, after trying all the relevant rules, the answer still has not been discovered, the program asks the user for the relevant information which will permit it to establish the validity of the premise clause. Thus, the rules unwind to produce a succession of goals (i.e., the system performs an exhaustive depth first search of the and/or goal tree formed by the rules), and it is the attempt to achieve each goal that drives the consultation.

The use of a rule-based representation of knowledge also makes it possible for the system to offer explanations of all intermediate and final conclusions. During the course of interaction with the user, for example, the consultation program requests many pieces of information. If the motivation for requesting any particular piece of information is unclear, the user may temporarily put off answering and instead inquire why such information is needed and how it will be used. The explanation program which allows him to do this can explore past, current, and future (potential) lines of reasoning, at varying levels of detail according to instructions from the user.

A second form of explanation is available after the consultation has ended. For example,again using MYCIN to illustrate, if a user asks "How did you determine the identity of the organism?" the program answers by displaying the rules which were actually used, and explaining, if requested, how each of the premises of the rules was established. This is something which someone familiar with the domain (in this case, a physician) can readily understand without having to know very much about either computers in general or this particular system. Note that it also provides a far more comprehensible and acceptable explanation that would be possible if the methodology employed were to be based statistical approaches to decision making.

The knowledge acquisition program can aid in the task of adding new "chunks" of knowledge to the program's expanding knowledge base. The rule-based representation of knowledge means that the expert himself can specify the new knowledge by expressing it in this format. He can thus help make the program more competent, without having to know anything about computer programming. In addition, since the rules are designed to be independent of one another, and are used by the program as necessary in order to deal with the particular situation under consideration, the addition of a new rule or modification of an existing rule does not require alteration of other items in the knowledge base, as is often necessary with systems using other methodologies.

2.5 Judgmental and Inexact Knowledge

Since we want to deal with real-world domains in which reasoning is often judgmental and inexact, we require some mechanism for being able to say that "A suggests B", or "C and D tend to rule out E." The numbers used to indicate the strength of a rule have been called Certainty Factors (CFs). The methods for combining CFs are embodied in a model of approximate implication. These are derived from and are related to probabilities, but are distinctly different. For the rulegiven above, the evidence is "suggestive" (.6 out of 1), but not absolutely certain. Evidence confirming an hypothesis is collected separately from that which disconfirms it, and the truth of the hypothesis at any time is the algebraic sum of the current evidence for and against it. This is an important aspect of the truth model, since it makes plausible the simultaneous existence of evidence in favor and against the same hypothesis. We believe this is an important characteristic of any model of inexact reasoning.

2.6 Incomplete knowledge

Another fundamental characteristic of the information in real world problems is its incompleteness. In a military domain for instance, information about an unfolding situation may be incomplete sometimes because it is totally unavailable, or more typically, because it is not possible to collect all necessary idata before decidin upon a course of action. Since both of these are common to problems in a wide range domains, a basic requirement of an intelligent system is that it be able to reach a decision based on whatever information is currently available.

Unlike many previous methodologies for decision making (e.g., decision trees),the goal-directed chaining ofthat we used above as our example rules is quite flexible with respect to the amount of information present, because the reasoning chains are constructed dynamically. Decision constraints (decision "logic") are computed at time of execution. There is thus no need to attempt to create a pre-determined sequence of decisions that takes account of every contingency. Instead the information present at the time of the problem-solving interaction will itself determine the direction in which reasoning ought to proceed.

2.7 Multiple Uses of Knowledge Bases

The accumulation of a large store of task-specific knowledge presents a number of unique opportunities. First, of course, it supports high performance, and this is typically why

it is assembled. But there are numerous other ways in which the same knowledge base can be employed, with little or no extra overhead. One area of investigation is its use as a reference source for computer-assisted education. That is, in much the same way that an expert taught the system about the domain, is it not possible to have the system teach a student? In a similar vein, it might be used as an on-line reference source, one that was easily searched, read, and modified as knowledge in the domain accumulated.

2.8 Summary of Advantages of Production Rule Methodology

2.8.1 Modularity and Ease of Modification by Non-programmer User:

As discussed above, the individual rules, as "units" of knowledge, can be removed or replaced quickly by the user-analyst. This makes for a highly adaptive man-computer system, one which gives the user the (correct) feeling that he is in charge of the nature and course of the analysis method and strategy. Such a system is bound to be more powerful and responsive than a computer-only inference system.

2.8.2 Flexibility with Robustness--Knowledge is separated from software

Most large programs are currently similar to a house of cards in that changes or additions must be made with the utmost care, lest unanticipated remote effects make the entire structure collapse. Generally there are too many unknown, unstable, and critical interdependencies. Using production rules, the effects of changes are propagated naturally and non-destructively.

2.8.3 Highly integrated and dynamic documentation

Documentation is natural, self-evident, and (where necessary) enforced. The rules themselves are represented in a language "natural" to the user-analyst (e.g. English for a physician-user of MYCIN, chemical graphs for a user of DENDRAL). The system can answer the primary explanatory questions concerning use of rules: HOW is the rule being used and WHY was the rule invoked in the line-of-reasoning. Rule forms govern the acquisition of a new rule, and include all information appropriate and useful for documentation (including ancillary information about when, by whom, and why it was added to the knowledge base). In addition, the same kind of "natural" questioning can be used to interrogate the knowledge base. Thus a Manual for the knowledge base is unnecessary.

2.8.4 Rule criticism during Acquisition

Because the acquisition of new rules is controlled by the so-called "rule models" inferred from an examination of the knowledge base, the system can catch many errors of commission and omission made by the user.

2.9 Technical Need

The defense systems of the United States are as much information processing systems as they are systems of weapons technology and use. Military personnel, particularly those in command and other decision-making positions, are routinely inundated with massive amounts of problem data. They use their expertise and good judgment to convert that data into relevant information on the basis of which intelligent choices and plans can be made. There is a pressing and omnipresent need in the services for computer-based intelligent assistance in this vital data-to-information transformation. Our basic research has been aimed at developing the methodology that will facilitate the programming of such agents; and at implementing real systems in physical,chemical,biological,medical,and military problem domains so as to give credibility to the view that such agents are possible.

Traditional mathematical methods are often powerful, but are sometimes abused, and in any event are scarce. These methods are, for most problems of interest, being pressed to their limits in appropriately representing problems of interest. Most of the world's (and the military's) problems are not basically problems of quantitative formulation and analysis, and the attempt to put them in this straightjacket is often frustrating.

Most real-world problems involving decision making and problem solving are symbolic in nature, involving qualitative reasoning, using knowledge that is informal, experiential, judgmental, and often tacit (i.e. private). As we have said, our research is aimed at this class of problems.

"Expert" knowledge-based intelligent-agent programs meet these kinds of needs (among many others):

a. the problem-dependent knowledge base can be made thorough and more complete than that known to any individual user

b. the use of the knowledge in decision-making and problem solving can be made highly systematic, so that relevant knowledge or appropriate solution paths are not inadvertently ignored.

c. the systems can be made available in crisis situations

that are typically characterized by information overload of the human analysts and decision makers.

d. symbolic knowledge can help to structure and guide expensive computational processes, by focusing the processing on appropriate calculations, thereby reducing the computational time and cost by (sometimes) orders of magnitude (this advantage was most obvious in the HASP domain).

Finally, from the point of view of the people doing analysis, decision-making, planning, etc. (i.e. from the human engineering point of view) there are these needs:

a. for systems that are easily expanded and modified, so that the system can change as human understanding of the problem changes

b. for programs that can represent human expertise and inexactness of human judgment in a natural way.

c. for programs that can build and maintain credibility with their users by offering clearly understandable explanations of their lines-of-reasoning,thereby making the users feel comfortable in accepting assistance from the agent.

2.10 Technology Transfer

In view of the importance of the needs described above, our project scientists have been, and are now, engaged in efforts to transfer our techniques and methods to technologists working on defense systems, and to various scientific disciplines.

Our previous proposals have documented our technology efforts of past years. The two year period just ending has been rich with such activities.

a. Feigenbaum and other project scientists have interacted vigorously with RAND Corporation scientists on the transfer of methods and systems developed here to the RITA system development for command-and-control applications.

b. We have had similar high levels of interaction with scientists at SRI on the development of their systems and methods in military applications. Of particular note is our interaction with them on Inference Nets; and the participation of Feigenbaum and Lederberg at a special meeting relating to EW analysis issues.

c. Feigenbaum and Nii have worked with the scientists at Systems Control Inc. on the follow-on effort to HASP (the SIAP program)>

d. Feigenbaum gave an all-day seminar to Senior Engineers at the National Security Agency on Knowledge-based systems, and our project scientists have held meetings with NSA scientists on this subject.

e. Feigenbaum is consulting with System Development Corporation on the writing of a tutorial report on Knowledge-based Systems for a military agency.

f. Feigenbaum has interacted with scientists at NELC on the formulation of a Navy project applying knowledge-based system methods to a C-C Testbed application.

g. Feigenbaum and project scientists have interacted with engineers at General Motors Research on the application of knowledge-based systems to their industrial problems.

h. Considerable interaction with industrial chemists regarding the use of the DENDRAL programs has been underway for some time. One of our scientists will be spending a year bringing up an "export" version of DENDRAL in England under the sponsorship of the Science Research Council and Shell Oil Co. Other industrial users of the programs include Eli Lilly Co., Monsanto Chemical Co.,Du Pont, and Kodak.

i. A general and massive effort at technology transfer has been the initiation of the Handbook of Artificial Intelligence, being written by Feigenbaum and his students (see subsequent proposal material for more details).

j. The AGE project described later in the proposal is a major effort to make available significant portions of our technology in the form of software packages to a broad scientific and engineering community.

3 ACCOMPLISHMENTS

The Heuristic Programming Project has become known as one of the most active artificial intelligence research groups in the country because of our long history of accomplishments as well as the intensity of our current work. Summarized below are the most notable accomplishments in the past two year period, followed by a review of our work in the years prior to that.

3.1 1976 to 1977

1) Production-Rules

The production rule methodology was extended and generalized in two important respects. First, the formalism for the premise clauses was extended to allow comparisons among objects as well as tests on individual objects. For example, instead of merely testing to see if the identity of an infecting organism is e.coli, the MYCIN program can now test to see if the current organism has the same identity as any other organism cultured from this patient. Second, explanation capabilities for production rule systems were extended to answer a broader range of questions. For example, questions about why the MYCIN program did NOT consider e.coli to be a likely infecting organism can now be answered. Negative questions are often highly relevant, but are difficult for a program to answer because their answers come from rules that the program did NOT invoke or procedures that did NOT get executed.

2) Planning...particularly planning of experiments

The framework was laid for a rule-based hierarchical planning program to guide scientific experimentation. The knowledge needed to make the planning inferences can come from multiple sources of scientific knowledge. (Coordinate funding to support much of this work was obtained from NSF.) The testbed for these ideas is the MOLGEN experiment planning program in molecular genetics, largely funded by the NSF. The planning problems are general: how to represent planning strategies, how to represent objects at arbitrary levels of detail, how to cope with imprecision in the planning rules.

3) Meta-DENDRAL: new results=new science

Meta-DENDRAL successfully formulated rules of mass spectrometry that were new to the science. These rules, along with a discussion of the methodology, were published in the scientific literature [[ref XXII}. The program was tested to see if it could rediscover the rules of mass spectrometry for two classes of chemical compounds that were already well understood (amines and estrogenic steroids). Then it was applied to three classes of compounds whose mass spectrometry was not as well known (mono-, di-, and tri-ketoandrostanes). The program produced three sets of rules that explained much of the significant data for these classes. The time for manual rule formation for these data was estimated to be several months. Automatic rule-formation was completed in less than an hour of processing time.

4) Meta-DENDRAL: generalization

Progress was made on generalizing the Meta-DENDRAL program ; and rules for a new domain were successfully discovered by the program. A scientific paper on this application was submitted for publication [[ref Schwenzer}. The new application was:

learning rules for interpreting signals from C13-NMR spectroscopy. The instrument produces data points in a bar graph in response to the resonance of each carbon-13 nucleus in the sample. The rules describe an environment of a C13 atom and predict a resonating frequency range for every atom that matches the description. The Meta-DENDRAL program needed some modification because the rules are predicting ranges of data points, and not precise processes, as for the mass spectrometry version.

5) Meta-DENDRAL: Methodology

A report was written on the AI methodology underlying Meta-DENDRAL [[ref BGB/TM}. The major idea is using knowledge of the domain to guide a learning program. The major difference between Meta-DENDRAL and statistical learning programs is that Meta-DENDRAL uses a strong model of mass spectrometry, including any assumptions the user cares to make about the domain, to guide the formation of explanatory rules.

6) Meta-DENDRAL: negative results

Our attempt to produce rules of various forms from Meta-DENDRAL was premature because the syntactic form of the rules is intimately tied to the rule generator. We are working on the problem of generalizing Meta-DENDRAL and this task is among those that we consider important. Of course, the program can search an immense variety of rules within a specified family. At issue here is the lack of flexibility and generality across family boundaries.

7) HASP: generalization

The control structure and representational scheme of the HASP program was successfully mapped into a new domain (image analysis of x-ray crystallographic data, as described in the background section). A working paper on this was published [[ref Nii}.

8) Knowledge Engineering: techniques for interactive control

Mechanisms were developed for giving users of rule-based interactive programs more control over the running programs. While a knowledge-based system is making its inferences, in between statements that get printed, a user often wonders what is going on and whether the answer from the program will be worth the wait. We introduced into the CONGEN program two helpful interrupt capabilities that can be generalized to other programs. First, the user can interrupt the computation to ask for an estimate of the amount of work remaining to be done. This requires a model of how to extrapolate from current status to

final status. Second, the user can request any, or all, of the intermediate solutions already generated in order to decide whether to continue the work or to introduce new constraints on the problem.

9) Rule-based teaching program: held in abeyance

Before we could produce an "intelligent teacher" that used an existing production rule knowledge for instructional purposes, it was necessary to work on the problem of providing explanations in many different ways (proposed here). Thus the effort expended on this problem only uncovered new research problems for the time.

10) Distributed computing

The first version of the "contract nets" programming organization was developed and a paper about it written. Contract nets will allow various types of AI methods to be programmed for parallel, asynchronous nets of computers.

11) Documentation of AI concepts, techniques, programming methodology: AI Handbook

Organization of the volume was completed, and first drafts of most of the volume were completed. The publication of a technical report was premature, but informal copies of material were sent to ARPA-IPTO.

3.2 1975 to 1976

1) The methodology and a prototype system for encoding rules of strategy for problem solving were developed for production rule systems. We call these "meta-rules" and have reported their use in [[ref Davis1976}. A meta-rule reorders or prunes the list of relevant domain rules to be invoked on the basis of the encoded strategy to pursue the consultation one way rather than another.

2) A program (named Teiresias) was written for interactive acquisition of new rules, using strong syntactic models of the rules to guide the interaction. This work was reported in [[ref Davis1976}. The system has strong expectations of what to expect in rules that make certain conclusions and in rules that refer to various attributes in their premises. Thus it can remind the user to include additional clauses in the rule and it can interpret many ambiguous statements written by the user.

3) A program was written to formulate models of rule-forms in a production rule system's knowledge base. These we call

"rule schemas" and their automatic formulation and use are reported in [[ref Davis1976}. These models are induced over the set of the existing rules in the system at any moment. They are changed each time a new rule is added so that the information is always current.

4) The MYCIN program was "emptied" of its medical knowledge to produce a program that could accept knowledge of non-medical (or other) domains. This version still makes strong assumptions about the organization of objects in the domain and about the nature of rules. If the structure of the new domain is similar to MYCIN's in these respects, however, the program can accept new rules and offer expert advice in this domain. One test domain was the diagnosis and recommendation for fixing problems in an automobile horn.

5) The RULEMOD program for "fine-tuning" Meta-DENDRAL's newly-discovered rules was finished. This program provides a number of important subtasks, including merging similar rules, making rules more specific or more general, and filtering out the weakest rules. RULEMOD checks for counterexamples to rules and uses this information in all of the named tasks. Because of the expense of computing counterexamples to possible rules, this computation is delayed until Meta-DENDRAL has a set of plausible rules, rather than computing counterexamples on each possible rule examined in the search of the rule space.

6) The RULEGEN component of Meta-DENDRAL was demonstrated to work with heuristic search methods used in many other AI programs (cf. Nilsson's book, Problem-solving Methods in Artificial Intelligence, McGraw-Hill). Guidance from a model of mass spectrometry is an important feature of RULEGEN. Also, the program uses problem data for pruning possible rules (and all more specific rules formed from those). The amount of data examined during the search is very large and the space of rules is immense, so the search needs to be rather coarse in order to produce plausible, but not necessarily optimal, rules.

7) A program was written to discover new, interesting conjectures in mathematics. This program, called AM, starts with a set of fundamental propositions of number theory and explores interesting extensions to the set, for example by looking at analogies and by looking for interesting properties that various subsets of the numbers share. This work was reported in [[Lenat}.

8) A rule-based system for integrating multiple sources of knowlege, in the task of interpreting 3-D image data, was designed and partly implemented. (This task was funded in large part by the National Science Foundation.) The data for this prototype system come from the domain of protein crystallography. KSs are being developed and implemented for identifying

individual molecules in the crystal structure, locating heavy atoms and cofactors, locating the backbone of the polymer, identifying side chains, matching the features in the data to the amino acid sequence, etc. The inferred structure is a multi-leveled hypothesis, conforming to the levels of detail to which model builders refer in applying their skill. A flexible, rule-based control structure allows the system to be exercised in many different modes, e.g. event-driven or goal-driven. Knowledge sources are kept distinct from the program that deploys them, so that the effectiveness of specific KSs on the system's performance can be readily investigated.

3.3 Years prior to 1975

Under ARPA support, and later under NIH support, the HPP developed a high performance, knowledge-based program for the interpretation and explanation of signal data from a mass spectrometer. This is the Heuristic DENDRAL program, which is now widely acclaimed as one of the most powerful and useful of AI programs.

The problem solving paradigm that has emerged from DENDRAL work is the so-called "plan-generate-test" paradigm. It is based on heuristic search of a space of possible hypotheses with planning before generation of hypotheses and testing of each generated candidate.

The generator for DENDRAL is a general-purpose graph generator which produces a list of all possible graphs containing specified numbers of nodes of various types. The most important features of the generator are that the list of graphs is guaranteed to be complete and non-redundant and, equally important, that the list need not be exhaustively generated. The generator can be constrained to produce only graphs that meet specified criteria that are inferred from the initial problem data.

The planning program is domain-specific. It contains a large amount of knowledge of mass spectrometer fragmentation processes, nuclear magnetic resonance phenomena, chemical stability, etc., and can accept additional knowledge (or assumptions) about the problem area from the scientist using the program.

The testing program uses a theory of mass spectrometry to make testable predictions for each candidate hypothesis. Matching the observed data with the prediction, then, allows the program to disconfirm some hypotheses and confirm the rest to varying degrees. The confirmed hypotheses are ranked and the most plausible hypotheses are given to the user as the best explanations of the initial data.

The success of any reasoning program is strongly dependent on the amount of domain-specific knowledge it contains. This is now almost universally accepted within AI, but DENDRAL's success is one of the reasons for its acceptance. We also showed the overall benefits of separating the knowledge of the domain from the computer code that manipulates that knowledge for problem solving.

Because of the difficulty of extracting specific knowledge from experts to put into the program, we began to explore the problems of transferring knowledge into a program efficiently. We have looked at two alternatives to "hand-crafting" each new knowledge base: interactive knowledge transfer programs and automatic theory formation programs. In this enterprise the separation of domain-specific knowledge from the computer programs themselves has been a critical component of our success.

We first developed a representation of knowledge that is natural and easy for experts to think about. The ideas had been discussed by others (e.g., Newell), but we brought them together in a problem-solving system in science. We codify individual facts about the domain as attributes and values associated with objects -- "the X of Y is Z". In addition, we developed an easy way of encoding and manipulating the degree of certainty (called the "certainty factor" or CF) associated with each fact. The rules of inference between facts are then encoded in conditional sentences, or production rules, which allow new conclusions to be made (with some CF) any time the statements mentioned in the premise clauses are true (or "true enough").

For interactive knowledge transfer we developed programs that carry on a dialogue with an expert in order to help him encode his knowledge of the domain. At first these programs were extremely simple, but we have by now given them the ability to understand stylized (not entirely free form) English sentences typed by the expert and have given them some understanding of the content, as well as the form, of the rules.

Automatic knowledge acquisition is a longer-term research problem. One of the stumbling blocks with the interactive knowledge transfer programs is that for some domains there are no experts with enough specific knowledge to make a high performance problem solving program. We were looking for ways to avoid forcing an expert to focus on original data in order to codify the rules explaining those data because that is such a time-consuming process. Therefore we began working on an automatic rule formation program (called Meta-DENDRAL) that examines the original data itself in order to discover the inference rules for that part of the domain.

The problem solving paradigm for Meta-DENDRAL is also the plan-generate-test paradigm used in Heuristic DENDRAL. In this

case the program generates plausible rules within syntactic and semantic constraints and within desired limits of evidential support. The model used to guide the generation of rules is particularly important since the space is enormous. The planning part of the program collects and summarizes the evidential support. The testing part looks for counterexamples to rules and makes modifications to the rules in order to increase their generality and simplicity and to decrease the total number of rules.

Another application of artificial intelligence has been made by the MYCIN group, now called the Knowledge-Based Consultation Systems group to reflect our interest in basic methodological issues. Work on MYCIN began in 1971 with a small group looking specifically at medical decision making. The resulting program, based on many of the ideas developed during work on DENDRAL, was shown to perform at the level of an expert in one area of medicine. The methodology was general, however, and we sought ARPA support in 1975 to extend that methodology beyond the one applications area. This was the start of our work on generalized production systems, along with programs that could explain the system's reasoning and programs to acquire new production rules from experts. Under separate funding the MYCIN program continues to develop into a high performance system that can provide expert advice about problems that are too complex for a non-expert to deal with alone.

Other research groups have studied MYCIN for their own applications, including ARPA-sponsored groups at SRI and RAND. The conceptual simplicity of the production rule interpreter in MYCIN makes it relatively easy to map the program into new domains, with their own sets of domain-specific rules. MYCIN's rule base contains about 400 rules at present and so we are looking closely at ways to keep the amount of processessing from growing with the number of rules.

4 PROPOSED WORK

4.1 Prologue

The Heuristic Programming Project's contract support from ARPA is part of a mosaic of funding that supports basic research in the design and engineering of knowledge-based systems. The work is usually set in the context of domains of knowledge in science or medicine. The ARPA contract supports this work in the direction of research goals of interest to ARPA's IPTO in the area of knowledge-based systems. Grant support from the National

Science Foundation and the National Institutes of Health provides domain expertise as well as some of the basic computer science support. because the task domains and the basic research goals of the ARPA-sponsored research mesh so well with NIH's goals for the SUMEX-AIM computer resource, pdp-10 time and file space is provided by SUMEX-AIM to the ARPA project. The synergism among these various projects and interests has been very great, making this project one of the most effective--and indeed one of the most cost-effective--in the history of artificial intelligence research.

4.2 Major Themes

The statement of tasks that follows is woven of three major themes:

I-A. Generalization of knowledge-based systems design and implementation techniques. The systems we have built are sufficient in number and variety to support an attempt to generalize our techniques so that they will be readily accessible to a broad community of system builders, not merely the small Stanford group. We believe that we are at a significant point of cumulation in this science, and that the product of the scientific generalization should be programs embodying the generalized methods serving future knowledge-based systems design and implementation efforts.

I-B. Extension of current research toward new or more powerful techniques for knowledge-based systems. The line of research we have been pursuing has been very fruitful. Most of the important scientific thrusts are currently active but not yet fully exploited. There are substantial scientific and practical benefits to pushing ahead to a fuller exploitation during the coming period. Much of the work on knowledge acquisition, representation of production rules, integration of multiple sources of knowledge,etc.,is to be viewed in this light.

II. Orientation toward other ARPA-IPTO advanced R&D efforts. We are orienting a portion of our total effort to a rule- based approach to some of the problems in the interpretation of image data; and to a study of the program organizations and signal-interpretation methods that might be used in distributed sensor net applications.

4.3 Tasks

The following list summarizes the research tasks to be performed during the two year period. It is broken into two major parts. The first set of tasks are frontier problems of

advanced symbolic computing. The second set includes research problems that relate directly to other ARPA-IPTO programs.

I. Expert Knowledge Based Systems and Heuristic Rule Technique Development

Proposed is a continuation of our basic research on knowledge-based systems design and techniques for representing "expert" symbolic knowledge in the form of heuristic rules. The research is composed of five major parts (A-E) discussed below.

A. Development of program-packages embodying generalized techniques for work on knowledge-based systems ; preparation of a compendium of these techniques and associated methods and programs in artificial intelligence research.

1. AGE package ("Attempt to GEneralize"). Isolate inference, control and representation techniques from previous knowledge-based programs; reprogram them for domain independence; write a rule-based interface that will help a user understand what the package offers and how to use the modules; and make the package available to other ARPA contractors, service labs doing knowledge-based systems development, and the general scientific community.

Detailed Discussion: The goal of this new effort is to construct a computer program to facilitate the building of knowledge-based systems. The design and implementation of the program will be based primarily on the experience gained in building knowledge-based systems at the Heuristic Programming Project in the last decade. The programs that have been built are: DENDRAL, meta-DENDRAL, MYCIN, AM, HASP, and MOLGEN (currently under development). Initially, The AGE program will embody methods used in our programs. However, the long-range objective is to integrate methods and techniques developed at other A.I. laboratories. The final product is to be a collection of useful "building-block" subprograms, combined with a knowledge-based front-end that will assist a user in constructing knowledge-based programs. It is hoped that AGE can speed up this process and facilitate transfer of the technology by: (1) packaging common AI software tools so that they do not need to be reprogrammed for every problem; and (2) helping people who are not knowledge-engineering specialists to write knowledge-based programs.

Two Specific Research Activities of the AGE Effort are:

1. The isolation of techniques used in knowledge-based systems.

It has always been difficult to determine if a particular problem-solving method used in a knowledge-based program is

"special" to a particular domain or whether it generalizes easily to other domains. In the currently existing knowledge-based programs the domain-specific knowledge and the manipulation of such knowledge using AI techniques are often so closely coupled that it is difficult to make use of the programs for other domains. We need to isolate the AI techniques that are general to determine precisely the conditions for their use.

2. Guiding users in the application of these techniques.

Once the various techniques are isolated and programmed for use, an "intelligent front end" is needed to guide users in their application. Initially, we assume that the user understands AI techniques and knows what he wants to do, but that he does not understand how to use the AGE program to accomplish his task. The program at this stage of the development will need to have the basic tools coupled with a package to guide the user in applying these tools. A longer-range interest involves helping the user determine what techniques are applicable to his task. That is, we assume that the user does not understand the necessary techniques of writing knowledge-based programs. Some questions to be posed are: What are the criteria for determining if a particular application is suited to a particular problem-solving framework? How do you decide the best way to represent knowledge for a given problem? There are some smaller, but by no means trivial, questions which also need answering. Is there a "best way" to write production rules which would apply to many task domains? Is there a data representation that would cover many tasks? What is the best way to handle differences in the ability of the users of the AGE program?

Research Plan: The AGE program will be developed along two separate fronts, both of which are divided into incremental development stages. The first of these fronts is the development of the ability to help build many different types of knowledge-based programs (the "generality" front). The second front is the development of "intelligence" in the interaction between the user and the AGE program; i.e. moving from dialogues on "how to use the tools in AGE" to "what tolls to use" (the "how-to-what" dialogue front). The current plan for the development of the AGE program, with approximate milestone dates, are listed below.

a. Generality: The development of a program package that will enable the user to build HASP-like knowledge-based programs. The applicable tasks are characterized by the need for: the integration of multiple sources of knowledge, multi-level representation of solution hypothesis, opportunistic problem-solving methods, and explanation capability of the reasoning steps. HASP-like paradigm has been used to solve problems of interpreting large amounts of digitized physical signals, but can also be extended to problems of processing large amounts of symbolic data. Dialogue: The development of dialogue to show the

user how to utilize the packaged components in AGE to build HASP-like programs. The interactive capability will be limited to: specifying how to build multi-level hypothesis structure; how to write production rules to represent domain knowledge; and how to use various techniques available for opportunistic hypothesis formation. (12/77)

b. Generality: Supplement the ability to build HASP-like programs with a capability to build MYCIN-like goal oriented programs. Dialogue: Same level of dialogue capability with additional ability to discuss how to chain rules and how to specify the necessary parameters for the context tree. (9/78) c. Generality: Same level as for 9/78, i.e. ability to build HASP-like, MYCIN-like or combination of HASP- and MYCIN-like knowledge-based programs. Dialogue: Begin to extract from the user some key charateristics of the task, and using that information begin to suggest appropriate knowledge representation and problem-solving techniques for the user's task. This interactive capability will be limited to the generality level at this point in the AGE development. (12/78) d. Test phase: the usefulness of the AGE system by developing an application program in some task domain. (a) An application program will be chosen from among ongoing program development efforts within our own project or within the ARPA contractor community. An application will be chosen whose primary task is that of interpreting large amounts of symbolic data or described signal data. Among the possibly appropriate tasks would be one of rule-based image analysis (see proposal section IIa.). (The search for an application will be concurrent with the development of the dialogue capabilities of 3., 12/78). (b) Collect specific knowledge needed for the application program and begin to develop the program using the AGE system (6/79).

2. PLAN package. Oriented toward the representation of plans-of-action and toward an expert's knowledge of the best problem solving strategies to employ in his domain. Will do inference on components of planning and strategy rules so that new plans and strategies can be constructed readily from previous ones. The representation will allow the manipulation of various "levels of detail" of plans and strategies. The package will be made available as previously mentioned in connection with AGE.

Detailed Discussion:

Before starting a technical presentation of the ideas for the Plan Package, it is worth highlighting some of the issues which motivate its development.

a. How can a variety of types of domain actions be accommodated in a knowledge base?

b. How can a variety of types of strategy and control knowledge be incorporated in a knowledge base?

c. How can a variety of types of problem solving states be expressed and manipulated by the system?

d. How should plans be represented?

e. How can the problem statements for a variety of types of problems be acquired?

f. How does the expression and representation of problem solving states relate to the expression of the domain and strategy knowledge?

The Plan Package consists of two major entities -- the Planning Network and the Strategy Package. The Planning Network is a set of software which manages the representation of the plans created during the problem solving process. When a problem is acquired from a user, it is represented as an initial planning network. Problem solving takes place as the active strategy rules manipulate the planning network to create solutions. The Strategy Package itself is discussed in the next section.

Since the planning state knowledge is important for the expression of strategy in the Plan Package, it is worthwhile exploring briefly the nature of this knowledge. It is useful to consider the planning network as being composed of three parallel planes -- the solution plane, the planning plane, and the focus plane. These planes contain (1) the solution steps (domain rule applications) and world states, (2) the planning and design steps and (3) the focus of attention knowledge respectively. All three planes of the network are built dynamically during the problem solving process. Different types of nodes in the network correspond to the different components of the problem solving process.

A number of issues have been raised about the management of strategy knowledge.

a. How should strategies be expressed?

b. How can strategy information be assimilated so that the system will use it appropriately when designing or explaining solutions?

c. How can a knowledge based system assist a domain expert in structuring and expressing his ideas about strategy?

Means-ends analysis is one of the simplest ideas in the current stock of methods for problem solving. As such, it should exist as a standard strategy in strategy package of artificial intelligence techniques to be used as needed. The current state

of artificial intelligence, where a researcher must re-code Means-ends analysis any time he wishes to use it is akin to a carpenter forging a new hammer for each job.

One approach for making an instance of Means-ends analysis available as a tool would be to provide a packaged program which accepts arguments for the various components of Means-ends analysis (eg. a difference table, difference function, etc.). The alternative being proposed here is a system which uses schemata to drive the strategy acquisition process and which can guide a user through the details. The goal is to create a supportive environment for the painless testing of fairly high level strategies. Such a system should be able to draw on its knowledge base to provide assistance in casting a problem into a Means-ends framework.

In summary, other systems have stumbled over the expression of more complex forms of domain and strategy rules and have been limited to solving a single kind of problem. We propose extending this work by developing what we have termed the Plan Package. The Plan Package consists of two major components - a schema-based representation for the problem-solving states termed the Planning Network and a schema-based representation for domain rules and strategies termed the Strategy Package. The Planning Network will provide a representation for a variety of types of problem solving so that the problem solving system will be able to solve more than one type of problem. The Strategy Package will provide a set of standard artificial intelligence strategies in the form of schemata, which may be instantiated into strategy rules when they are supplied with the particulars of domain knowledge. These schemata will facilitate the acquisition of tailored strategies by guiding a user a step at a time through the particulars of the acquisition process.

Research Plan:

The Plan Package will be developed and tested in the domain of molecular genetics as part of the MOLGEN project. It will be further developed and extended to other domains as a test for generality as part of the AGE project.

Time	Milestone
3mo MORSE	-Molgen Object Rule & Strategy Editor operational objects.
6mo MORSE	operational for rules. System capable of simulating several kinds of laboratory experiments. Geneticists begin using system to build Knowledge Base.
9mo MORSE	operational for some strategies. Capable of planning limited classes of experiments. Primitive performance monitoring operational. Begin CS experiments.
12mo MOLGEN	starts having impact by trying to design real experiments. Penny starts getting involved to use parts of system for experiments in AGE project.